2d-mapless navigation based on point cloud observations using actor-critic agent

Emanuel Muñoz
Department of Electrical Engineering, UTEC
emanuel.munoz@utec.edu.pe

Elmer Escandon
Department of Electrical Engineering, UTEC
elmer.escandon@utec.edu.pe

Abstract

This work proposes an end-to-end learning approach for navigation applications, using 2D raw LiDAR measurements and relative positions from the goal. A novel network structure is applied to characterize the 2D measurements in a pair fashion to train an actor-critic model. Through a simple ideal environment training, the results indicate convergence with high rewards to reach its position.

1 Introduction

Indoor autonomous navigation is a very active field for being useful in applications on warehouse, social, and emergency robotics. Traditional approaches for autonomous navigation require several modules and ad hoc solutions for its environment. Modular approach could be avoided using learning techniques such as Deep Reinforcement Learning (Deep RL). In Zhelo et al. [2018], the work proposes an agent based on curiosity to navigate using only raw sensor data. Similarly, in Leiva and Ruiz-del Solar [2020], the authors present several architectures based on DDPG for mapless navigation. In this work, we propose a end-to-end approach based on an actor-critic agent. Alike previous works, we do not depend on a mapping stage, instead 2D raw points from a LiDAR sensor are applied. Also, this work presents a novel network architecture to deal with 2D-point cloud measurements. 2D-points require additional treatment due to some reasons mentioned below. The agent is implemented in a simulated 2D robot environment.

2 Methodology

2.1 Problem Formulation

The agent-environment emulates a mobile robot without orientation navigating through a 2D space at each time step t. In each state s_t, the robot observes o_t, executes action a_t according to a policy $\pi(a_t|o_t)$, calculates a reward scalar value r_t, and transitions to a new state s_{t+1}. The robot's actions are controlled in a discrete fashion, where only a certain direction must be chosen (up, down, left, right) with a specific velocity of $\{-2, 2\}ms^{-1}$. Additionally, the agent's observations are defined by o_{lidar} and o_{target}. The term o_{lidar} represents Cartesian (x_i, y_i) pairs ($i \in \{1, .., m\}$), obtained from the LiDAR measurements, while the $o_{\text{odom}} = \mathbf{p}^{x,y}_{\text{robot}} - \mathbf{p}^{x,y}_{\text{goal}}$ indicates the distance to the goal position in Cartesian coordinates from the robot's perspective.

Furthermore the reward function considers three cases per each time step t. For that, a threshold value ($\epsilon = 0.4m$) is compared to the robot-goal distance defined as $\rho_{\text{target}} = ||o_{\text{odom}}||_2$. First case, ρ_{target} is greater or equal than ϵ, the reward r^t_{reach} with value of 10 is applied. Similarly, the r^t_{nav} reward is chosen when ρ_{target} is less than ϵ. The r^t_{nav} reward guides a proximity indicator to the goal, defined as $r^t_{\text{nav}} = K(\rho_{\text{target}})$, where K is a normalization value set to -0.1. Finally, the $r^t_{\text{collision}}$ reward is used with a value of -10 in case the robot crashes with an obstacle.

This paper proposes certain assumptions to adequately characterize the sensor values in our learning structure. Firstly, the NaN values provided by the sensor are changed by the maximum values possible, as a consequence the number of measurements (m) remains constant in each iteration. Secondly, the measurements are treated in pairs to provide further insight in the learning model, since those indicate possible obstacle representation instead of solely distances.

2.2 Model and agent

Obtaining features from a 2D point cloud turns into a challenge for several reasons: sample size, adequate learning, and ordering as it is discussed by Qi et al. [2017]. In this work, we propose a new architecture to manage this issue with some considerations for o_{lidar}. In fig. 1, we present our architecture for training considering an A2C agent with two heads: critic and actor values. We first diminished the problem of the sample size through fixing the number of measurements as explained previously. Then, for an adequate learning, we considered that the Cartesian pairs from o_{lidar} must interact explicitly. We defined several mini layers for every pair (x_i, y_i) so then they could be concatenated in an MLP network. This section is named Local Notion so it provides information about the surroundings of the robot. Having the points in order to feed the neural network is also a problem, so previous to this step we added a shuffle layer.

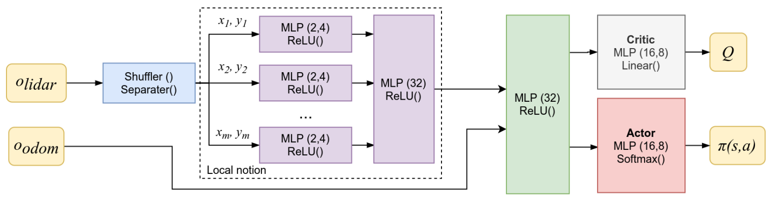

Figure 1: Scheme of actor-critic architecture. Observations o_{lidar} and o_{odom} are separated inputs.

3 Results

We implemented this architecture based on PyTorch and OpenAI Gym tools. Open code is available [1]. In fig. 2a, we present an own simulated environment used for being faster for testing and evaluation. It considers random obstacle generation and collisions in simplified scenario.

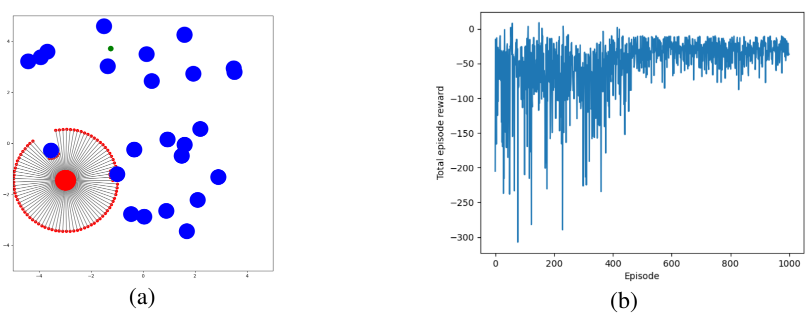

Figure 2: (a) The simulated environment: a robot in red with surrounded lines representing the sensor, obstacles in blue, and target in green. (b) Accumulated reward for each episode. Total episodes: 1000. A2C trained with $lr = 10^{-4}$ and $\gamma = 0.999$

We ran the simulation and architecture in an Intel® Core™ i7-8750H CPU @ 2.20GHz. We ran 1000 episodes in ~ 2 hours. Time varies depending on the success ratio of the robot during training. In Fig. 2b, we show the accumulated reward for each episode. An explicit convergence to obtain high rewards is shown. Simulated robot works well considering accurate measurements.

4 Conclusions

2D-LiDAR observation allows a mapless navigation in an unstructured environment using deep reinforcement learning and the presented architecture. We also presented a solution for dealing with 2D-pointcloud for neural network feeding using a novel architecture and some considerations. Future work could consider a more realistic scenario and a more complex environment.

[1] *github.com/EmanuelSamir/2d-navigation-drl*

References

F. Leiva and J. Ruiz-del Solar. Robust rl-based map-less local planning: Using 2d point clouds as observations. *IEEE Robotics and Automation Letters*, 5(4):5787–5794, 2020.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

O. Zhelo, J. Zhang, L. Tai, M. Liu, and W. Burgard. Curiosity-driven exploration for mapless navigation with deep reinforcement learning. *arXiv preprint arXiv:1804.00456*, 2018.